HTTP status codes for the top queries [Packetbeat] ECS

Top 10 HTTP requests [Packetbeat] ECS

Network Traffic Between Hosts [Packetbeat Flows] ECS

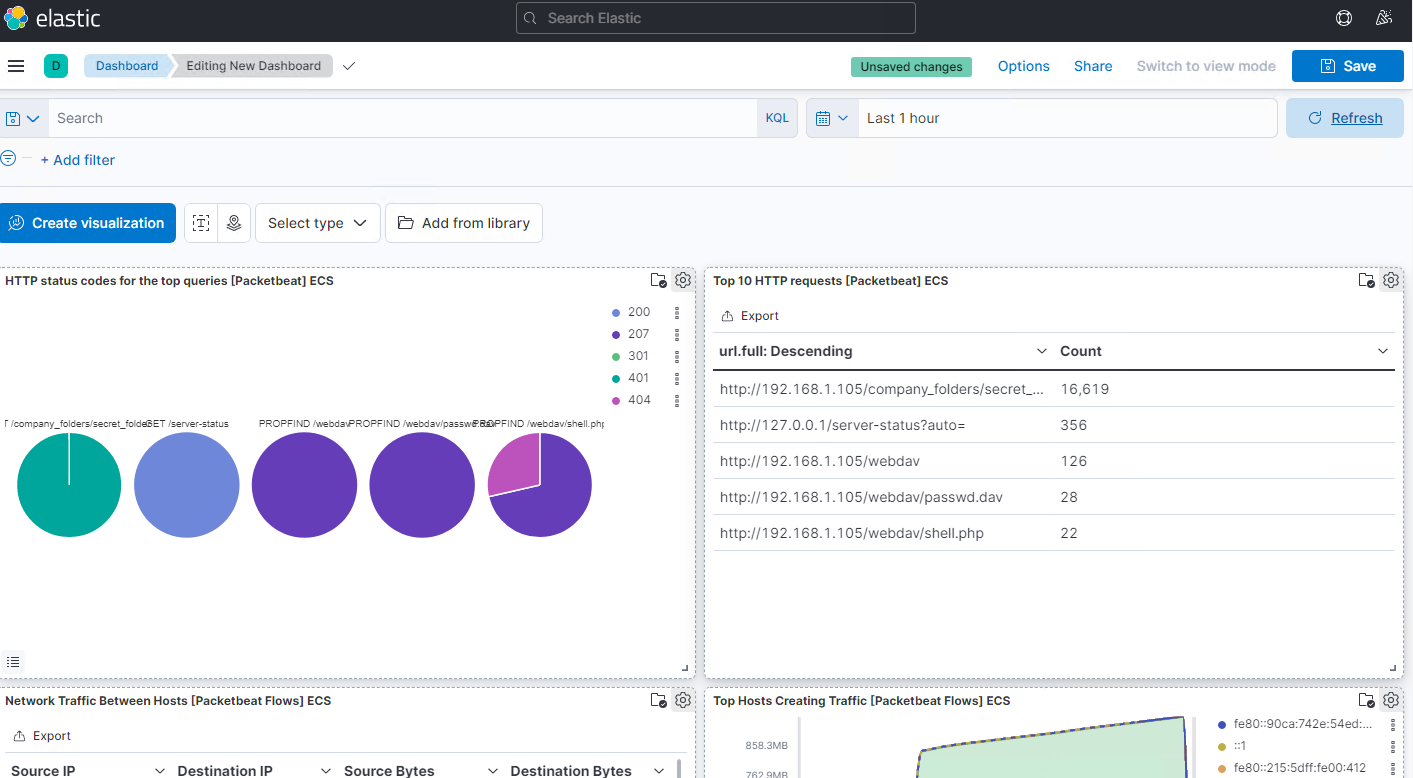
Top Hosts Creating Traffic [Packetbeat Flows] ECS

Connections over time [Packetbeat Flows] ECS

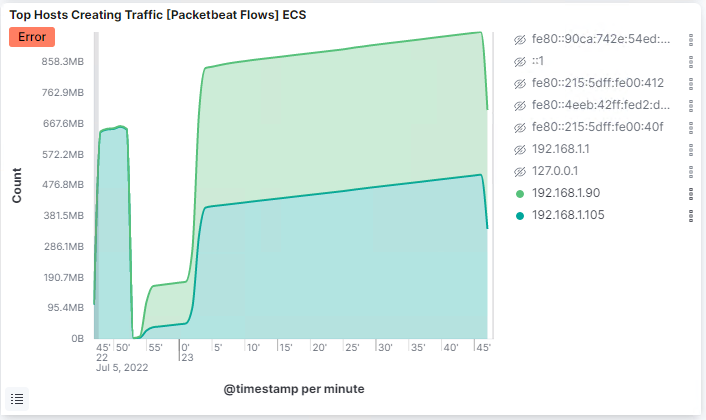
HTTP error codes [Packetbeat] ECS

Errors vs successful transactions [Packetbeat] ECS

HTTP Transactions [Packetbeat] ECS



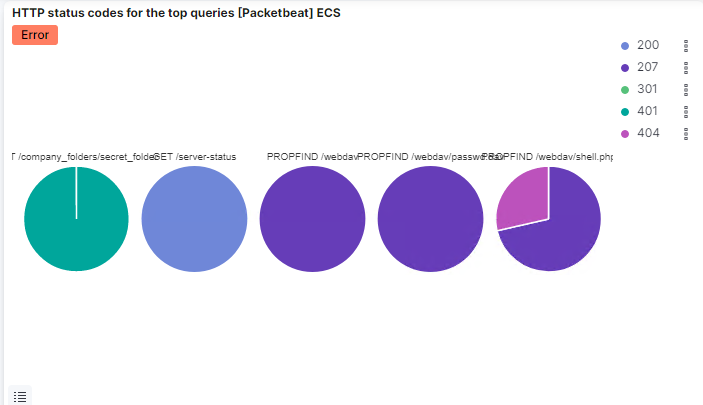
1. Identify the offensive traffic.  
   * Identify the traffic between your machine and the web machine:
     + When did the interaction occur?



Time is 23:05

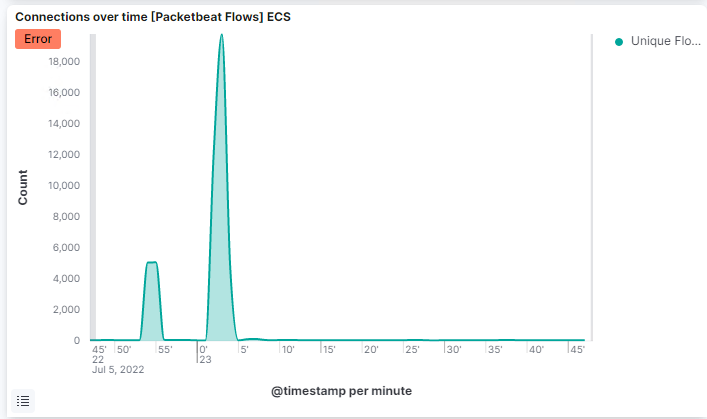
* + - What responses did the victim send back?

We can see 401, 301, 207, 404 and 200 as the top responses.

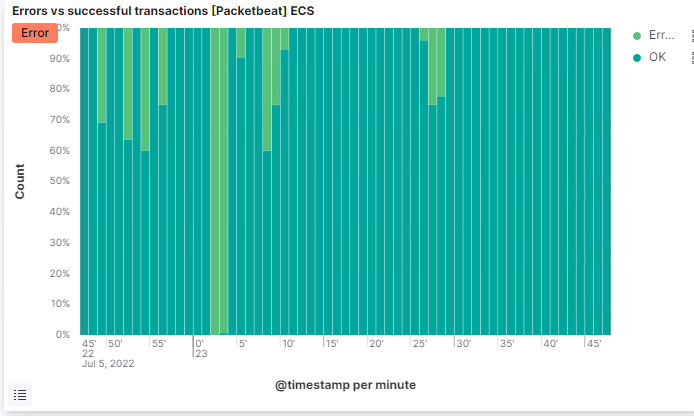


* + - What data is concerning from the Blue Team perspective?

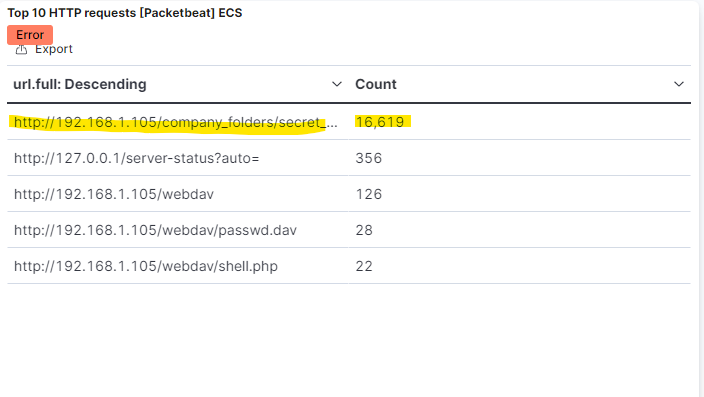
We can see a connection spike in the Connections over time [Packetbeat Flows] ECS



We can also see a spike in errors in the Errors vs successful transactions [Packetbet] ECS



1. Find the request for the hidden directory.  
   * In your attack, you found a secret folder. Let's look at that interaction between these two machines.
     + How many requests were made to this directory? At what time and from which IP address(es)?



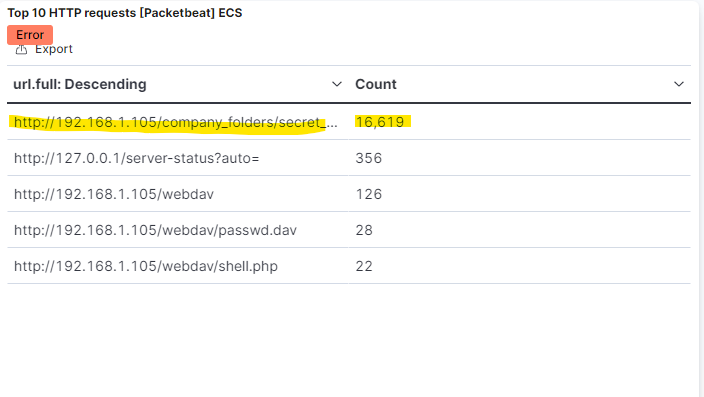
* + - Which files were requested? What information did they contain?
    - What kind of alarm would you set to detect this behavior in the future?

We could set an alert that goes off for any machine that attempts to access this directory or file.

* + - Identify at least one way to harden the vulnerable machine that would mitigate this attack.

This directory and file should be removed from the server all together.

1. Identify the brute force attack.  
   * After identifying the hidden directory, you used Hydra to brute-force the target server. Answer the following questions:
     + Can you identify packets specifically from Hydra?

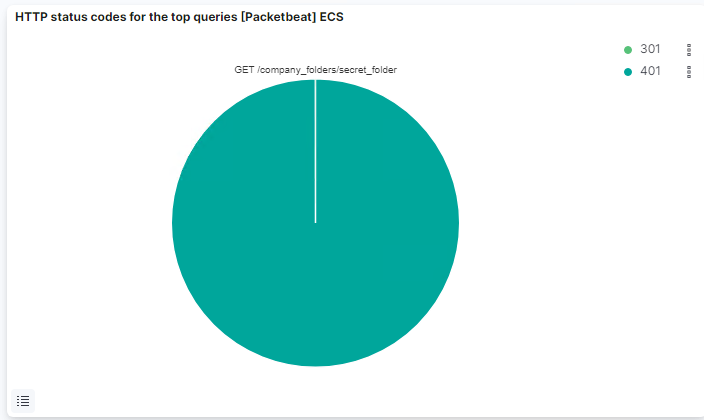


Because there are so many in a short time frame

* + - How many requests were made in the brute-force attack?

16,619

* + - How many requests had the attacker made before discovering the correct password in this one?



1 successful request

* + - What kind of alarm would you set to detect this behavior in the future and at what threshold(s)?

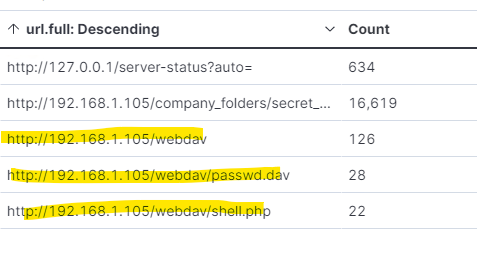
We could set an alert if 401 Unauthorized is returned from any server over a certain threshold that would weed out forgotten passwords. Start with 10 in one hour and refine from there.

We could also create an alert if the user\_agent.original value includes Hydra in the name.

* + - Identify at least one way to harden the vulnerable machine that would mitigate this attack.

After the limit of 10 401 Unauthorized codes have been returned from a server, that server can automatically drop traffic from the offending IP address for a period of 1 hour. We could also display a lockout message and lock the page from login for a temporary period of time from that user.

1. Find the WebDav connection.  
   * Use your dashboard to answer the following questions:
     + How many requests were made to this directory?



* + - Which file(s) were requested?

Passwd.dav and shell.php

* + - What kind of alarm would you set to detect such access in the future?

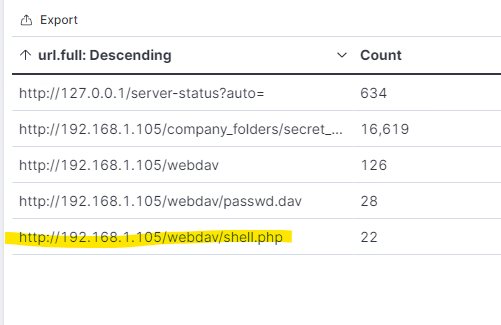
We can create an alert anytime this directory is accessed by a machine *other* than the machine that should have access.

* + - Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Connections to this shared folder should not be accessible from the web interface.

Connections to this shared folder could be restricted by machine with a firewall rule.

1. Identify the reverse shell and meterpreter traffic.  
   * To finish off the attack, you uploaded a PHP reverse shell and started a meterpreter shell session. Answer the following questions:
     + Can you identify traffic from the meterpreter session?



I can tell bc it has the word shell in it

* + - What kinds of alarms would you set to detect this behavior in the future?

We can set an alert for any traffic moving over port 4444.  
We can set an alert for any .php file that is uploaded to a server.

* + - Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Removing the ability to upload files to this directory over the web interface would take care of this issue.